1. **Elimination** In this problem we’ll work through the mechanics of using ELIMINATE for inference in a directed graphical model. Consider the directed graphical model below over the binary variables W, X, Y, and Z. Let \( P(W = 0) = 0.5 \). All the local conditional probabilities are the same with the following values: \( P(X = 0|W = 0) = 0.6 \), \( P(X = 0|W = 1) = 0.2 \). We will use ELIMINATE to compute \( P(W = 0|Z = 1) \).

(a) Write out the summations involved in carrying out the elimination algorithm to find \( p(w, z) \) and \( p(z) \), as in (3.8) to (3.15) in Chapter 3 of the text. Use the elimination order \( (Z, Y, X, W) \).

(b) Using ELIMINATE (with \( \overline{z} = 1 \)), write down (as tables) the resulting intermediate terms, and \( p(w, \overline{z}) \).

(c) Compute \( P(W = 0|Z = 1) \).

2. **Belief Propagation** Consider an undirected tree, \( T = (V, E) \), where \( V \) are the nodes and \( E \) are the edges. We can use the belief propagation algorithm to compute all the single node marginals \( p(x_i) \). You are to provide a modification of the belief propagation algorithm that will yield the edge marginals; i.e., the probabilities \( p(x_i, x_j) \) for \( (i, j) \in E \). What is the running time of the algorithm?

3. **Factor Graph** Consider the factor graph shown below, where circles represent binary random variables and squares represent factors. \( X_1, X_2, X_3 \) are message bits that we are trying to estimate; \( X_4 \) and \( X_5 \) are odd parity check bits, i.e. they are true if an odd number of their parents are true, otherwise false. In our model, we have \( X_4 = X_1 \oplus X_2 \), and \( X_5 = X_2 \oplus X_3 \), where \( \oplus \) represents exclusive or. Suppose we receive noisy observations \( Y_{1:5} \) of all five bits. Let \( F(s, i) = p(X_i = s|y_i) \) be the local evidence vector at node \( i \). Thus \( F(:, i) = |1, 0| \) means that \( X_i \) is observed to be in

![Factor Graph](image)

**Figure 1:** Error Correcting Factor Graph
state \( s = 1 \) (true), \( F(\cdot; i) = [0, 1] \) means \( X_i \) in \( s = 2 \) (false) and \( F(\cdot; i) = [0.5, 0.5] \) is uninformative. Note that I use state 1 and 2 to represent the true and false value because I cannot use 0 to refer to MATLAB’s matrix entries. The posterior is thus

\[
P(X_{1:5}|y_{1:5}) = \frac{1}{Z} g(X_1, X_2, X_4) \cdot g(X_2, X_3, X_5) \cdot \prod_{i=1}^{5} F(X_i, i)
\]

Your task is write a MATLAB program to decode the message by computing \( P(X_{1:3}|y_{1:5}) \) and the normalization constant \( Z \). Let’s see an example.

If the local evidence is as follows

\[
F = \begin{bmatrix}
0.5 & 0.5 & 0 & 0 & 0 \\
0.5 & 0.5 & 0.5 & 1 & 1
\end{bmatrix};
\]

then all the message bits are uncertain, but both parity checks are perfectly observed to be in state 2 (false). Your MATLAB program should compute the value of \( Z \) and the distribution \( P(X_{1:3}|y_{1:3}) \) as follows:

\[
\begin{array}{cccc}
X_1 & X_2 & X_3 & \text{Prob} \\
1 & 1 & 1 & 0.5 \\
0 & 1 & 1 & 0.0 \\
1 & 0 & 1 & 0.0 \\
0 & 0 & 1 & 0.0 \\
1 & 1 & 0 & 0.0 \\
0 & 1 & 0 & 0.0 \\
1 & 0 & 0 & 0.0 \\
0 & 0 & 0 & 0.5 \\
\hline
\text{Z} & & & 0.25 \\
\end{array}
\]

Submit your MATLAB code and compute a similar table for each of the following four scenarios:

\[
\begin{align*}
\text{Scenario 1: } F &= [0.9 \ 0.5 \ 0.5 \ 0 \ 0; \ 0.1 \ 0.5 \ 0.5 \ 1 \ 1] \\
\text{Scenario 2: } F &= [0.9 \ 0.9 \ 0.9 \ 0 \ 0; \ 0.1 \ 0.1 \ 0.1 \ 1 \ 1] \\
\text{Scenario 3: } F &= [0.9 \ 0.9 \ 0 \ 0 \ 0; \ 0.1 \ 0.1 \ 1 \ 1] \\
\text{Scenario 4: } F &= [0.9 \ 1 \ 0 \ 0 \ 0; \ 0.1 \ 0 \ 1 \ 1 \ 1]
\end{align*}
\]

Explain your answer in scenario 4.

4. **Bayesian estimation of Gaussian parameters** We consider a Bayesian approach for learning the parameters of a Gaussian distribution. Let \( x_1, \ldots, x_n \) be \( n \) IID samples from \( \mathcal{N}(m, \frac{1}{r}) \), where \( m \) and \( r \) are the mean and the precision, which is defined as the reciprocal of the variance. The normal-gamma prior, \( P(m, r|\alpha, \beta, \mu, \tau) \), is defined as:

\[
p(m, r|\alpha, \beta, \mu, \tau) = p(r) \cdot p(m|r) \\
= \frac{\beta^\alpha}{\Gamma(\alpha)} r^{\alpha-1} e^{-\beta r} \cdot \sqrt{\frac{r}{2\pi}} e^{-\frac{r}{2\pi} (m-\mu)^2}
\]
In the normal-gamma prior, the precision $r$ has a Gamma$(\alpha, \beta)$ distribution and the conditional distribution of the mean $m$ is $N(\mu, \frac{1}{r})$.

Show that using the normal-gamma prior, the posterior is also a normal-gamma density with parameters $\hat{\alpha}, \hat{\beta}, \hat{\mu}$ and $\hat{\tau}$

$$
\hat{\alpha} = \alpha + \frac{n}{2} \\
\hat{\beta} = \beta + \frac{n - 1}{2} S_x + \frac{n \tau (\bar{x} - \mu)^2}{2(\tau + n)} \\
\hat{\mu} = \frac{\tau \mu + n \bar{x}}{\tau + n} \\
\hat{\tau} = \tau + n
$$

where $\bar{x}$ is the sample mean and $S_x$ is the sample variance.

5. **Bayesian Model comparison** We are asked to build a suspicious object recognition system for the airport. The problem can be formulated as follows: Let $D$ be the set of all possible objects that can pass through our system. $D_c \in D$ is a small set of sample suspicious objects $x_1, \ldots, x_N$ manually identified for training. Let $x \in U \setminus D$, we would like to compute how likely that $x$ is an suspicious object.

One possible approach is to see how likely we could describe $x$ and $D$ using the same set of parameters. In Figure 2, the left graphical model $M_1$ describe the scenario where both $x$ and $D_c$ are governed by the same parameter $\theta$ while the right model $M_2$ shows that they are governed by different parameters. To compare these two models, we can use the **Bayes Factor**, which is similar to the likelihood ratio in hypothesis testing:

$$
\text{BayesFactor} \triangleq \frac{p(M_1|x, D_c)}{p(M_2|x, D_c)}
$$

A Bayes Factor larger than one will indicate that $x$ is more likely to be a suspicious object.

Figure 2: Two Models describing the data

(a) Assume $P(M_1) = P(M_2)$, show that we can compute the Bayes Factor as follows:

$$
\text{BayesFactor} = \frac{P(x|D_c)}{P(x)P(D_c)}
$$
where

\[
P(x|D_c) = \int P(x|\theta)P(\theta|D_c)d\theta
\]

\[
P(x) = \int P(x|\theta) = \int P(x|\theta)P(\theta)d\theta
\]

\[
P(D_c) = \int \prod_{i=1}^{n} P(x_i|\theta)P(\theta)d\theta
\]

Since \( P(D_c) \) does not depend on \( x \), we can redefine our test as follows:

\[
\text{score}(x) \triangleq \frac{P(x|D_c)}{P(x)} \geq P(D_c)
\]

(b) Assume each item \( x_i \in D_c \) is a binary vector \( x_i = (x_{i1}, \ldots, x_{ij}) \) where \( x_{ij} \in \{0, 1\} \), and that each element of \( x_i \) has an independent Bernoulli distribution:

\[
P(x_i|\theta) = \prod_{j=1}^{J} \theta_i^{x_{ij}} (1 - \theta_j)^{1-x_{ij}}
\]

The conjugate prior for the parameters of a Bernoulli distribution is the Beta distribution:

\[
P(\theta) = \prod_{i=1}^{J} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \theta_i^{\alpha_j-1}(1 - \theta_j)^{\beta_j-1}
\]

where \( \alpha = (\alpha_1, \ldots, \alpha_J) \) and \( \beta = (\beta_1, \ldots, \beta_J) \) are the hyper-parameters. Show that

\[
P(D_c) = \prod_{i=1}^{J} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \frac{\Gamma(\hat{\alpha}_j)\Gamma(\hat{\beta}_j)}{\Gamma(\hat{\alpha}_j + \hat{\beta}_j)}
\]

where \( \hat{\alpha}_i = \alpha_i + \sum_{i=1}^{N} x_{ii} \) and \( \hat{\beta}_i = \beta_i + N - \sum_{i=1}^{N} x_{ii} \).

(c) Let the unknown object \( x = (x_1, \ldots, x_J) \). Show that the logarithm of our score function admits a very simple computation as follows:

\[
\log(\text{score}(x)) = c + \sum_{j=1}^{J} q_j x_j
\]

where

\[
c = \sum_{j=1}^{J} \log(\alpha_j + \beta_j) - \log(\alpha_j + \beta_j + N) + \log \hat{\beta}_j - \log \beta_j
\]

\[
q_j = \log \hat{\alpha}_j - \log \alpha_j - \log \hat{\beta}_j + \log \beta_j
\]

Hint: Use the fact that \( \Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha) \) for \( \alpha > 0 \) and \( x_i \) can only be 0 or 1.